



public class OsamaBankAccount {

private int accountID;

private String customerName;

private double balance;

private static int counter=0;

public OsamaBankAccount(String customerName) {

counter++;

this.accountID =counter;

this.customerName = customerName;

balance=0;

}

public OsamaBankAccount(String customerName, double balance) {

counter++;

this.accountID = counter;

this.customerName = customerName;

this.balance = balance;

}

public void deposit(double amount){

else

balance-=amount; balance+=amount;

}

public void withdraw(double amount){

if(balance<amount)

System.out.println("The balance is not enough");

}

public void transfer(double amount,OsamaBankAccount out){

withdraw(amount);

out.deposit(amount);

}

public int getAccountID() {

return accountID;

}

public String getCustomerName() {

return customerName;

}

public double getBalance() {

return balance;

}

}

public class CheckingAccount extends OsamaBankAccount{

private final int free\_transactions=3;

private final double transaction\_fee=500;

private int transactionCount;

public CheckingAccount(String customerName) {

super(customerName);

this.transactionCount =0;

}

public CheckingAccount(String customerName, double balance) {

super(customerName, balance);

this.transactionCount =0;

}

@Override

public void withdraw(double amount) {

transactionCount++;

super.withdraw(amount);

}

void deductFees(){

if(transactionCount>free\_transactions){

double fees=transaction\_fee\*(transactionCount - free\_transactions);

super.withdraw(fees);

}

transactionCount=0;

}

}

public class SavingsAccount extends OsamaBankAccount{

private double interestRate;

public SavingsAccount(String customerName,double interestRate) {

super(customerName);

this.interestRate = interestRate;

}

public SavingsAccount(String customerName, double balance,double interestRate) {

super(customerName, balance);

this.interestRate = interestRate;

}

public void addInterest(){

double interst=(getBalance()\*interestRate);

super.deposit(interst);

}

public double getInterestRate() {

return interestRate;

}

}

import java.util.Scanner;

public class Osama\_International\_Bank {

public static void main(String[] args) {

OsamaBankAccount[] accounts = new OsamaBankAccount[10];

int numOfAccounts = 0;

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("الرجاء اختيار الخيار:");

System.out.println("1- إنشاء حساب");

System.out.println("2- إيداع");

System.out.println("3- سحب");

System.out.println("4- إضافة فائدة");

System.out.println("5- خصم الرسوم");

System.out.println("6- تحويل");

System.out.println("7- عرض معلومات الحساب");

System.out.println("8- الخروج");

int option = scanner.nextInt();

switch (option) {

case 1:

System.out.println("الرجاء اختيار نوع الحساب:");

System.out.println("1- حساب التوفير");

System.out.println("2- حساب الشيكات");

int accountType = scanner.nextInt();

System.out.println("الرجاء إدخال اسم العميل:");

String customerName = scanner.next();

System.out.println("الرجاء إدخال المبلغ:");

double initialBalance = scanner.nextDouble();

switch (accountType) {

case 1:

System.out.println("الرجاء إدخال معدل الفائدة:");

double interestRate = scanner.nextDouble();

accounts[numOfAccounts] = new SavingsAccount(customerName, initialBalance, interestRate);

numOfAccounts++;

System.out.println("تم إنشاء حساب التوفير بنجاح.");

break;

case 2:

accounts[numOfAccounts] = new CheckingAccount(customerName, initialBalance);

numOfAccounts++;

System.out.println("تم إنشاء حساب الشبكات بنجاح.");

break;

default:

System.out.println("نوع حساب غير صالح.");

break;

}

break;

case 2:

System.out.println("الرجاء إدخال رقم الحساب:");

int accountNumber = scanner.nextInt();

System.out.println("الرجاء إدخال المبلغ:");

double depositAmount = scanner.nextDouble();

OsamaBankAccount account = findAccount(accounts, numOfAccounts, accountNumber);

if (account != null) {

account.deposit(depositAmount);

System.out.println("تم إيداع المبلغ بنجاح.");

} else {

System.out.println("رقم الحساب غير صالح.");

}

break;

case 3:

System.out.println("الرجاء إدخال رقم الحساب:");

accountNumber = scanner.nextInt();

System.out.println("الرجاء إدخال المبلغ:");

double withdrawAmount = scanner.nextDouble();

account = findAccount(accounts, numOfAccounts, accountNumber);

if (account != null) {

account.withdraw(withdrawAmount);

System.out.println("تم سحب المبلغ بنجاح.");

} else {

System.out.println("رقم الحساب غير صالح.");

}

break;

case 4:

System.out.println("الرجاء إدخال رقم الحساب:");

accountNumber = scanner.nextInt();

account = findAccount(accounts, numOfAccounts, accountNumber);

if (account instanceof SavingsAccount) {

SavingsAccount savingsAccount = (SavingsAccount) account;

savingsAccount.addInterest();

System.out.println("تم إضافة الفائدة بنجاح.");

} else {

System.out.println("رقم الحساب غير صالح أو ليس حساب توفير.");

}

break;

case 5:

System.out.println("الرجاء إدخال رقم الحساب:");

accountNumber = scanner.nextInt();

account = findAccount(accounts, numOfAccounts, accountNumber);

if (account instanceof CheckingAccount) {

CheckingAccount checkingAccount = (CheckingAccount) account;

checkingAccount.deductFees();

System.out.println("تم خصم الرسوم بنجاح.");

} else {

System.out.println("رقم الحساب غير صالح أو ليس حساب شبكات.");

}

break;

case 6:

System.out.println("الرجاء إدخال رقم الحساب المرسل:");

int senderAccountNumber = scanner.nextInt();

System.out.println("الرجاء إدخال رقم الحساب المستلم:");

int receiverAccountNumber = scanner.nextInt();

System.out.println("الرجاء إدخال المبلغ:");

double transferAmount = scanner.nextDouble();

OsamaBankAccount senderAccount = findAccount(accounts, numOfAccounts, senderAccountNumber);

OsamaBankAccount receiverAccount = findAccount(accounts, numOfAccounts, receiverAccountNumber);

if (senderAccount != null && receiverAccount != null) {

senderAccount.transfer(transferAmount, receiverAccount);

System.out.println("تم التحويل بنجاح.");

} else {

System.out.println("رقم الحساب المرسل أو المستلم غير صالح.");

}

break;

case 7:

System.out.println("الرجاء إدخال رقم الحساب:");

accountNumber = scanner.nextInt();

account = findAccount(accounts, numOfAccounts, accountNumber);

if (account != null) {

System.out.println("معلومات الحساب:");

System.out.println("رقم الحساب: " + account.getAccountID());

System.out.println("اسم العميل: " + account.getCustomerName());

System.out.println("الرصيد: " + account.getBalance());

} else {

System.out.println("رقم الحساب غير صالح.");

}

Break;

case 8:

System.out.println("تم الخروج.");

System.exit(0);

default:

System.out.println("الرجاء اختيار خيار صالح.");

break;

}

}

}

private static OsamaBankAccount findAccount(OsamaBankAccount[] accounts, int numOfAccounts, int accountNumber) {

for (int i = 0; i < numOfAccounts; i++) {

if (accounts[i].getAccountID() == accountNumber) {

return accounts[i];

}

}

return null;

}

}